Astronomical Data Analysis Software and Systems VI
ASP Conference Series, Vol. 125, 1997
Gareth Hunt and H. E. Payne, eds.

ETOOLS: Tools for Photon Event Data

M. Abbott, T. Kilsdonk, C. Christian,[1] and E. Olson

Center for EUV Astrophysics, 2150 Kittredge Street, University of California, Berkeley, CA 94720-5030

M. Conroy, J. Herrero, and R. Brissenden

Smithsonian Astrophysical Observatory, 60 Garden Street, Cambridge, MA 02138

Abstract. We present an overview of the ETOOLS project and describe some of the design features of the ETOOLS system.

1. Goals

Our primary goal is to provide a set of generic tools for the basic manipulation of event data. We take the word generic to have several implications. The tools should be generic in function, meaning they are useful with any kind of event data and do not depend on details that may be specific to the telescope or instrument that produced the data. However, the tool set should be extensible to include new, possibly instrument-specific, tools.

The tools should also be generic in environment. By that, we mean that the tool set should function both stand-alone and within common data reduction environments, such as the IRAF CL or IDL.

Finally, the tools should not impose any particular format on the event data files, but, instead, be able to adapt to any format.

2. What is Event Data?

Event data refers to the very rich data sets produced by photon counting detectors such as those commonly used in high-energy telescopes. Some recent examples of missions that have produced, or will produce, event data are *Einstein, ROSAT, EUVE, ASCA, XTE*, and *AXAF*.

In event data, each event (such as a photon) on the detector is tagged with its arrival time, detector position, and other information. These tags are retained through the later processing of the events, so that a scientist analyzing the data can use filtering techniques to examine subsections of the complete data set, or form secondary data products such as images or light curves by binning the data in any desired manner.

Another typical feature of event data sets is the presence of auxiliary information from the telescope or the spacecraft on which it may be operating, in ad-

[1] also Space Telescope Science Institute, 3700 San Martin Dr., Baltimore, MD 21218

dition to the events themselves. The auxiliary data are most often time-ordered and tabular in nature, and are frequently used to determine the "quality" of the event data they accompany.

Because of this structure, event data lend themselves both to a very detailed and flexible analysis, in which examining small numbers of (or even individual) events can be useful, and to the more usual kinds of image analysis. Event data sets will typically consist of several event lists and tables of auxiliary information, along with various filters that may have been established, and images and light curves that have been computed.

3. The Scope of ETOOLS

Levels of abstraction are imposed on event data in order to analyze and understand it:

1. The data are stored in basic objects such as images and tables, which provide appropriate manipulations (e.g., filtering of tables, sectioning of images).
2. Relationships and interactions are supported between basic objects, which implement additional functionality needed to fully represent event data (see §4). For example, a "filter" object contains the time intervals to be used to filter a table of events, and is persistently associated with the event table.
3. Scientific meaning is attached to event data objects. For example, a particular filter object is identified as the "SAA removal filter," or a particular table is identified as containing a light curve of the source (a light-curve object).

ETOOLS addresses the intermediate, second level. We build upon existing libraries, which support basic objects, and provide applications that manipulate event data. But we leave the assignation of scientific meaning and manipulation of higher-level objects to tools (and possibly libraries), which are layered on top of ETOOLS.

The ETOOLS I/O library implements the Event Data Model in an application programming interface, which is then called from the ETOOLS applications. The I/O library calls on existing, basic object libraries.

4. The Event Data Model

We have developed a model for event data that describes the objects and operations needed for their analysis. An overview is given here (not all details are described).

Dataset. The dataset is an object that contains all of the component parts of an event data observation in a directory-like structure. It supports operations to add, remove, and list contained objects.

Property List. Every object in the dataset has an associated property list, which is a set of zero or more properties. A property is a string keyword and an associated value in one of several data types.

Image. Image objects are largely identical to well-known existing image objects such as FITS or IRAF images. Each is composed of pixel data and an associated WCS object. Operations will exist to read and write from the image or sections of it.

WCS. WCS objects encode the transformation between two coordinate systems. They are used in the data model in conjunction with image objects and the binning of tables.

Table. Table objects start with the capabilities of familiar table objects such as ST Tables or FITS Bintables. Additional supported features include a property list associated with each table column, binning of tables into images, and filtering.

Binning is supported by allowing table objects to be opened as image objects. Each table has a list of binning keys; each key indicates the columns defining the bins, the column to sum, and the WCS object to attach to the resulting image.

Filtering is supported as an integral part of reading tables. A list of filtersets is associated with a table; the active set is applied at the time of a read. Each filterset specifies the type of filter object to be applied and the table columns it applies to.

Interval Filter. An interval filter is a one-dimensional filter that specifies a set of valid ranges in a quantity. Interval filters are associated with tables through filtersets.

Region Filter. A region filter is a two-dimensional filter that defines a valid area in the space defined by two quantities. Region filters are associated with tables through filtersets.

5. The ETOOLS I/O Library

In order to achieve our goals of environment and file format independence, the ETOOLS I/O library is implemented in two layers: generic code and kernels. These layers provide a simple way to isolate the code, which depends on the details of a particular file format.

A kernel must be written for each underlying set of file formats and/or basic object libraries that will be used. Each kernel is responsible for implementing a set of functions—the kernel programming interface or KPI—which are called from the generic code. Kernels must implement missing functionality in the underlying libraries to provide the basic object capabilities assumed by the generic code.

To extend the ETOOLS library and applications to work with a new file format, only a new kernel is needed. See the paper on the implementation of the ASC Data Model (Herrero et al. 1997) in this volume for some examples of kernels under development.

The ETOOLS library and applications are written in ANSI C for maximal portability. Kernels may be implemented in any desired manner as long as the KPI is supported. Kernels may have dependencies on certain run-time environments. ETOOLS sites can select any desired kernels for installation. For example, a site that does not have IRAF installed can build ETOOLS by not including the IRAF-based kernels (perhaps working entirely with FITS).

6. ETOOLS Applications

A suite of non-interactive tools will perform all of the operations supported by the event data model. These are aimed primarily at scripting uses, but will also be suitable for users who prefer a command line interface.

A small number of interactive tools with graphical user interfaces will support the display and editing of datasets, tables, and interval filters. We have adopted SAOtng as the image display for ETOOLS.

Applications are written in ANSI C for portability. In order to function properly inside certain common data reduction environments, applications may contain a very thin "front end" of environment-specific code.

The ETOOLS applications will function in at least two environments: as a package of IRAF tasks and as a set of UNIX programs. Further environments can be supported with proper additions to the front end.

7. Project Status

The design of the library is completed and coding is ongoing. We are using a prototype kernel based on ST Tables and IRAF images to validate the software. Kernels based on FITS and on the IRAF QPOE file format are in planning and development stages at the ASC, separate from the ETOOLS effort.

Non-interactive tools that use the completed sections of the library have been developed. More will be added as the library progresses. Work is now concentrating on the interactive applications.

The ETOOLS project is planned for completion in early 1997. All products will be available from the Center for EUV Astrophysics and Smithsonian Astrophysical Observatory WWW and ftp sites, as well as other locations to be determined. Look for future announcements of ETOOLS beta and final releases on the ADASS newsgroups.

Acknowledgments. ETOOLS is supported by NASA contract NAS5-32698 through the Astrophysics Data Program. The Center for EUV Astrophysics is a division of the UC Berkeley Space Sciences Laboratory.

References

Herrero, J., Oberdorf, O., & Conroy, M. 1997, this volume, 469